Technical Cybersecurity

NMAP Scanning

Sniff & Scan

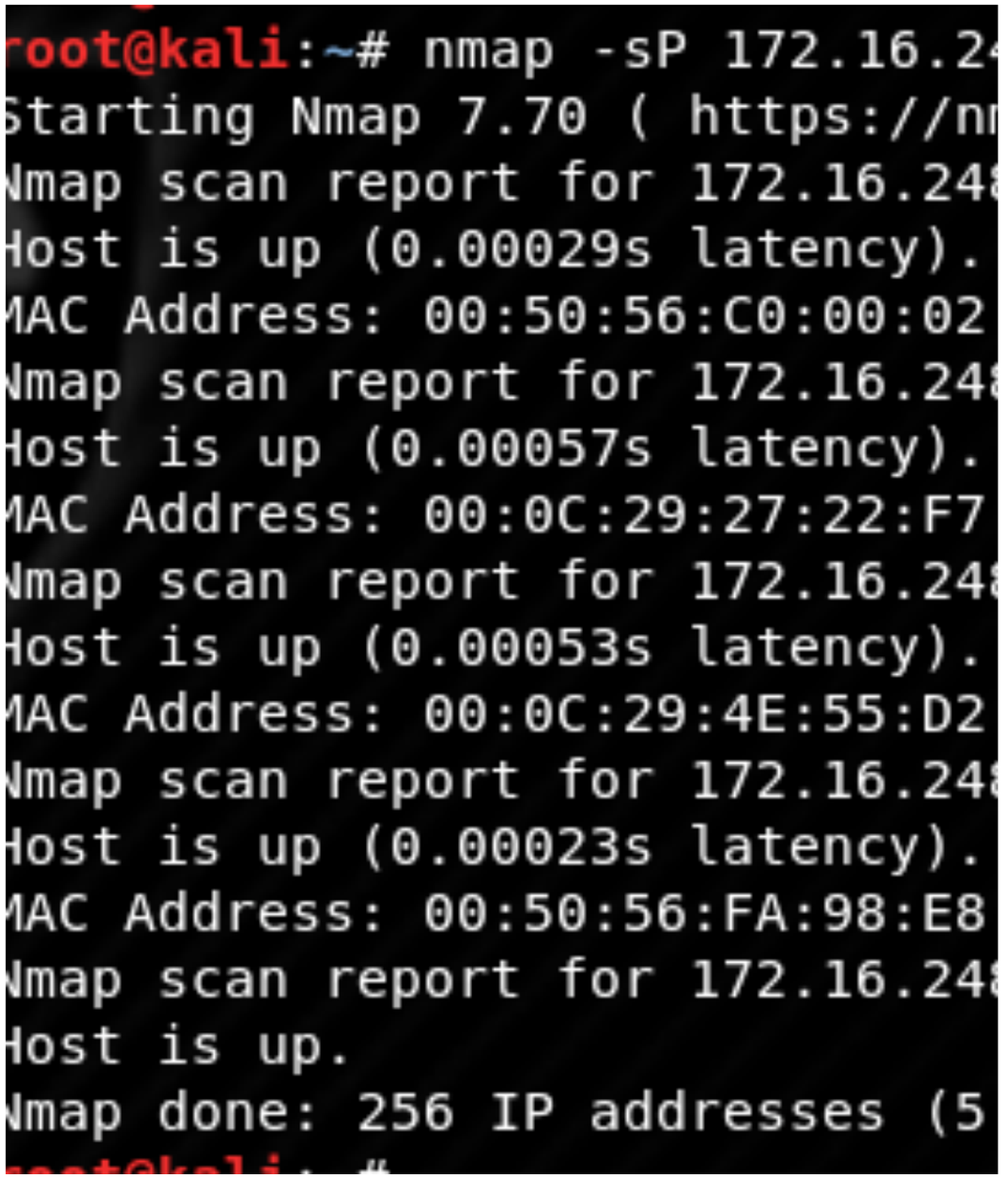
When scanning, you may want to collect network traffic

- You can go back and analyze what exactly happened during the scan
- You probably don't always want to do it, but it can be useful
  - masscan? probably not.
  - detailed nmap scan v. a single target? probably.

Discovery

172.16.248.0/24

- This is the subnet we're on (well, I'm on; yours might differ, but I'm going to use this in the examples)
- The 24 means that we're only interested in the first 24 bits of the address (i.e. 172.16.248.*)
- So what do we find?



First Scan

NMAP

- In one window, we'll run NMAP
- **$ nmap -oN out.txt -sP 172.16.248.***

TCPDUMP

- TCPDump in the other
- **$ tcpdump -nn -i eth1 -w scan.pcap net 172.16.248.0/24**

INTERACTIVE CONTROLS

- pressing d, v, or return will increate debug information, verbosity, and give you the status of the current scan, respectively
- press shift-{letter} to decrease debug info or verbosity

```
root@kali:~# nmap -oN out.txt -sP 172.16.248.*
Starting Nmap 7.70 ( https://nmap.org ) at 2018-1
Nmap scan report for 172.16.248.1
Host is up (0.00019s latency).
MAC Address: 00:50:56:C0:00:02 (VMware)
Nmap scan report for 172.16.248.128
Host is up (0.00034s latency).
MAC Address: 00:0C:29:27:22:F7 (VMware)
Nmap scan report for 172.16.248.129
Host is up (0.00029s latency).
MAC Address: 00:0C:29:4E:55:D2 (VMware)
Nmap scan report for 172.16.248.254
Host is up (0.00017s latency).
MAC Address: 00:50:56:FA:98:E8 (VMware)
Nmap scan report for 172.16.248.130
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned
```

Results

We have five hosts we can see.

ip.src == 172.16.248.130

No.	Time	Source	Destination	Protocol	Length	Info
515	1.980133	172.16.248.130	172.16.248.1	DNS	85	Standard query 0x66e8 PT
516	1.980295	172.16.248.130	172.16.248.1	DNS	87	Standard query 0x66e9 PT
517	1.980381	172.16.248.1	172.16.248.130	ICMP	70	Destination unreachable
518	1.980384	172.16.248.130	172.16.248.1	DNS	87	Standard query 0x66ea PT
519	1.980393	172.16.248.1	172.16.248.130	ICMP	70	Destination unreachable
520	1.980463	172.16.248.130	172.16.248.1	DNS	87	Standard query 0x66eb PT
521	1.980665	172.16.248.1	172.16.248.130	ICMP	70	Destination unreachable
522	1.980708	172.16.248.1	172.16.248.130	ICMP	70	Destination unreachable
527	5.981896	172.16.248.130	172.16.248.1	DNS	87	Standard query 0x66ec PT
528	5.981984	172.16.248.130	172.16.248.1	DNS	87	Standard query 0x66ed PT
529	5.982043	172.16.248.1	172.16.248.130	ICMP	70	Destination unreachable
530	5.982043	172.16.248.130	172.16.248.1	DNS	87	Standard query 0x66ee PT
531	5.982052	172.16.248.1	172.16.248.130	ICMP	70	Destination unreachable
532	5.982098	172.16.248.130	172.16.248.1	DNS	85	Standard query 0x66ef PT
533	5.982149	172.16.248.1	172.16.248.130	ICMP	70	Destination unreachable

```
▸ Frame 515: 85 bytes on wire (680 bits), 85 bytes captured (680 bits)
▸ Ethernet II, Src: Vmware_8f:a0:6e (00:0c:29:8f:a0:6e), Dst: Vmware_c0:00:02 (00:50:56:c0:00:02)
▸ Internet Protocol Version 4, Src: 172.16.248.130, Dst: 172.16.248.1
▸ User Datagram Protocol, Src Port: 45407, Dst Port: 53
▸ Domain Name System (query)
```

TCPDUMP

Load the PCAP file (scan.pcap)

What did we do?

NMAP

- We executed a probing NMAP scan.

TCPDUMP

- We captured the contents of the NMAP scan via TCPDUMP, saving the data to scan.pcap (PCAP means Packet CAPture).

WIRESHARK

- We used wireshark on our kali box to examine PCAP data
  - **$ wireshark**

More scans next!